1. UNCONDITIONAL OPTIMIZATION, MULTIVARIATE CASE

1.1. **The Newton method.**

Let $f_1, f_2, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ be multivariate differentiable real functions. Consider the following system of m equations:

$$\begin{aligned} f_1(x) &= 0 \\ f_2(x) &= 0 \\ &\cdots\cdots \\ f_m(x) &= 0 \end{aligned}$$

Approximate these multivariate functions with the well-known first degree Taylor polynomials at point $\mathbf{x}_k \in \mathbb{R}^n$, that is,

$$\begin{aligned} f_1(x) &\approx f_1(x_k) + \nabla f_1(x_k)(x - x_k) \\ f_2(x) &\approx f_2(x_k) + \nabla f_2(x_k)(x - x_k) \\ &\cdots\cdots \\ f_m(x) &\approx f_m(x_k) + \nabla f_m(x_k)(x - x_k) \end{aligned}$$

Let the components of the vector $F : \mathbb{R}^n \to \mathbb{R}^m$ be the multivariare functions $f_1(x), f_2(x), \ldots, f_m(x)$. Apply the Newton method to the system of equations $F(x) = 0$ ($F : \mathbb{R}^n \to \mathbb{R}^m$). Then you get

$$F(x) = F(x_k) + F'(x)(x - x_k),$$

where

$$F'(x) = \left[\frac{\partial f_i(x)}{\partial x_j}\right]_{i,j=1}^{m,n} \in \mathbb{R}^{m \times n}$$

is the Jacobian matrix. From this we obtain the approximation sequence

$$0 = F(x_k) + F'(x)(x_{k+1} - x_k),$$

from which the system of equations

$$F'(x)(x_{k+1} - x_k) = -F(x_k)$$

follows. If $m = n$, and $F'(x)$ a non-singular matrix, then

$$x_{k+1} = x_k - \left[F'(x_k)\right]^{-1} F(x_k) \qquad (k = 0, 1, \ldots).$$

1.2. **Algorithm.**

The previous formulas are used for the Newton method. But to avoid computing the inverse of the Jacobian matrix $F(x)$ it is preferable to use following iteration

(1) FOR $k = 0, 1, \ldots$

(2) $$F'(x_k)\, s_k = -F(x_k)$$

(3) $$x_{k+1} = x_k + s_k$$

If we apply the Newton method to the stationary system of equations $F(x) = \nabla f(x) = 0$, then the Jacobian matrix is of the form:

$$F'(x) = Hf(x) = Hf(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]_{i,j=1}^{n},$$

which is just the Hessian matrix of function f. The Newton method is used here in two ways. In the first case we do not invert the matrix. The iteration is then of the form:

(1) FOR $k = 0, 1, \ldots$

(2) $$Hf(x_k)\, s_k = -\nabla f(x_k)$$

(3) $$x_{k+1} = x_k + s_k$$

This will be called the **Newton's formula I.**

Using the inverse technique the approximating sequence is of the form:

(1) FOR $k = 0, 1, \ldots$

(2) $$s_k = (-Hf(x_k))^{-1}\, \nabla f(x_k)$$

(3) $$x_{k+1} = x_k + s_k$$

This will be referred to as the **Newton's formula II.**

1.3. **Remarks.**

Note that the matrix $Hf(x_k)$ is symmetric. If $x_k \approx x_{\min}$ and $Hf(x_{\min})$ is positive definite, then $Hf(x_k)$ is also positive definite.

The cost of the stepwise computation of the Newton method is the solution of a symmetric linear system of equations, which is the evaluation of gradient $\nabla f(x)$ and the $Hf(x)$ Hessian matrix.

The above Newton method can be obtained by means of geometric reasoning. Let us approximate function f in point x_k with the quadratic expression

$$f(x_k + s) \approx q_k(\delta) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T Hf(x_k)\, s.$$

Its minimum location is the solution of the linear system of equations $Hf(x_k)\, s = -\nabla f(x_k)$. From this minimum value of function f can be approximated by:

$$x_{k+1} = x_k - [Hf(x_k)]^{-1} \nabla f(x_k).$$

1.4. **Example.**

Exercise: Find the minimum of the the function

$$f(x, y) = 4x^2 + 4xy + 2y^2 - 10x - 12y + 2$$

by the Newton method, with guess vector $x_1 = [0, 0]$.

Solution: The gradient and the Hessian matrix are as follows:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 8x + 4y - 10 \\ 4x + 4y - 12 \end{bmatrix}, \qquad Hf(x) = \begin{bmatrix} 8 & 4 \\ 4 & 4 \end{bmatrix}.$$

Replacing in the argument x by x_1 the gradient and the Hessian matrix become:

$$\nabla f(x_1) = \begin{bmatrix} -10 \\ -12 \end{bmatrix}, \qquad Hf(x_1) = \begin{bmatrix} 8 & 4 \\ 4 & 4 \end{bmatrix}.$$

1.) Apply the Newton's formula I. Then the system of linear system of equations to solve is:

$$H(x_1)s_1 = -\nabla f(x_1)$$

or equivalently,

$$\begin{aligned} 8a + 4b &= 10 \\ 4a + 4b &= 12, \end{aligned}$$

where a and b are the respective coordinates of vector s_1. Its solution: $a = -1/2 = -0.5, b = 7/2 = 3.5$, and so the approximated value of x_2 is equal to

$$x_2 = x_1 + s_1 = \begin{bmatrix} -0.5 \\ 3.5 \end{bmatrix}.$$

We found that the optimum is obtained after one step. It is easy to check that this is actually the optimum point: $x_{\min} = [-0.5, 3.5]$. The resulting value of the objective function equals $f_{\min} = -33/2 = -16.5$.

2.) Apply the Newton's formula II. To this end compute the inverse of the Hessian matrix, which is as follows:

$$H^{-1}f(x) = \begin{bmatrix} 1/4 & -1/4 \\ -1/4 & 1/2 \end{bmatrix}$$

and apply formula

$$s_k = -H(x_k)^{-1}\nabla f(x_k)$$

to get

$$s_k = -\begin{bmatrix} 1/4 & -1/4 \\ -1/4 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} -10 \\ -12 \end{bmatrix} = \begin{bmatrix} -1/2 \\ 7/2 \end{bmatrix}$$

which leads to

$$x_2 = x_1 + s_1 = \begin{bmatrix} -0.5 \\ 3.5 \end{bmatrix}.$$

In the example, we found that the optimal solution is obtained in just one step. This is always the case when the objection function to be minimized is quadratic.

Remember in the second form of the Newton method when function f is replaced with its second degree Taylor polynomial, then both this and the function itself coincide which always the case for quadratic functions.

1.5. **Modified Newton's method.**

In the application of the Newton method it may happen that the Hessian matrix $Hf(x_k)$ is singular, or vector s_k is not a decreasing direction, or even if it a decreasing direction ($\nabla f(x_k)s_k < 0$), the function value at point x_{k+1} is not smaller that the function value at point x_k. The modified Newton method is conceived to by-pass these problems. (It can also be helpful to couple the Newton method to a linear optimization. This will also be shown later).

The modified Newton method is used in the following form:

FOR $k = 0, 1, \ldots$

$$[Hf(x_k) + E_k]\, s_k = -\nabla f(x_k)$$

$$x_{k+1} = x_k + s_k$$

where E_k is a matrix with non-negative diagonal such that $\nabla^2 f(x_k) + E_k$ is well-conditioned positive definite matrix.

Many algorithms exist for the choice of the diagonal matrix E_k. Here we shall deal with two such choice algorithms. The most spread **Gill and Murray iteration**, and the second one the so-called **Levenberg-Marquardt method**.

1.6. **The Gill-Murray algorithm.**

Let be given a symmetric matrix A, a number $\varepsilon \geq 0$ and a parameter $beta > 0$. The iteration generates an upper triangular matrix R and a diagonal matrix $E = diag_{(\varepsilon_i) \geq 0}$ such that $A + E = R^T R$.

FOR $i = 0, 1, \ldots, n$

$$\gamma_{ij} = a_{ij} - \sum_{k=1}^{i-1} r_{kj} r_{ki}, i \leq j \leq n$$

$$\mu_i = \max\{|\gamma_{ij}| : i < j \leq n\}$$

$$r_{ii} = \max\{\varepsilon, |\gamma_{ii}|^{1/2}, \tfrac{\mu_i}{\beta}\}$$

$$r_{ij} = \frac{\gamma_{ij}}{r_{ii}}, i < j \leq n$$

$$\varepsilon_i = r_{ii}^2 - \gamma_{ii}$$

The recommended value of parameter β:

$$\beta^2 = \max\left\{\frac{1}{n}\max\{|a_{ij}| \ : \ i \neq j\}, \max\{|a_{ii}| \ : \ 1 \leq i \leq n\}\right\}.$$

1.7. **The Levenberg-Marquardt method.**

We replace the Hessian matrix by a symmetric positive definite matrix B which is defined as follows:

$$\mathbf{B} = \mathbf{H}(\mathbf{x}_k) + E\varepsilon_k,$$

where $\varepsilon_k \geq 0$ is a parameter. Use the Newton method to solve the linear system of equations

$$Bs_k = (H(x_k) + E\varepsilon_k)s_k = -\nabla f(x_k)$$

for s_k. During the implementation of the algorithm, parameter
ε_k is replaced by the ratio

$$R_k = \frac{f(x_k) - f(x_{k+1})}{q(x_k) - q(x_{k+1})}.$$

Levenberg and Marquardt proposed the following procedure.
For $k = 1$: We start from an arbitrary vector $x_k \in \mathbb{R}^n$ and an arbitrary parameter $\varepsilon_k > 0$.
1. We try to compute the Cholesky decomposition of matrix $Hf(x_k) + E\varepsilon_k$. If it is not successful, then replace ε_k by ε_k . Continue this try together with the modification of the value of ε_k the Cholesky decomposition is successful.
2. If the Cholesky decomposition is successful, meaning that matrix $Hf(x_k) + E\varepsilon_k$ is positive definite, then solve the linear system of equations $(Hf(x_k) + E\varepsilon_k)s_k = -\nabla f(x_k)$ for s_k .
3. Compute the next iteration: $x_{k+1} = x_k + s_k$.
4. Stop if $\|x_{k+1} - x_k\| < \varepsilon$, otherwise do the following.
5. Compute the R_k.
If $0 < R_k < 0.25$, akkor legyen $\varepsilon_{k+1} = 4\varepsilon_k$.
If $R_k > 0.75$, tchoose $\varepsilon_{k+1} = \frac{1}{2}\varepsilon_k$.
If $0.25 \leq R_k \leq 0.75$, choose $\varepsilon_{k+1} = \varepsilon_k$.
If $R_k \leq 0$, choose $\varepsilon_{k+1} = 4\varepsilon_k$ and reset x_{k+1} to x_k , that is, $x_{k+1} := x_k$.
6. Continue the process, $k := k + 1$.

1.8. **The Quasi-Newton methods.**

The quasi-Newton methods are approximate Newton methods for which the stepwise computational cost (time) is extremely small and therefore the total computational cost (time) is less than the cost resulting from the Newton methods. The basic idea behind them is as follows.

Consider the non-linear system of equations $F(x) = 0$ $(F : \mathbb{R}^n \to \mathbb{R}^n)$. Set $x_0 \approx x^*, B_0 \approx F'(x^*)$. In the Newton method replace the Jacobian matrix $F'(x_k)$ by a suitably defined approximation matrix $B_k \approx D'(x_k)$:

(1) FOR $k = 0, 1, \ldots$
(2) $\quad B_k s_k = -F(x_k)$
(3) $\quad x_{k+1} = x_k + s_k$

Matrix B_k is derived from matrix B_{k-1} by adding to it one or more dyadic matrices. The rank-one modification of B_k:

$$B_k = B_{k-1} + u_k v_k^T \quad (u_k, v_k \in \mathbb{R}^n),$$

the rank-two modification of B_k:

$$B_k = B_{k-1} + u_k v_k^T + w_k z_k^T \quad (u_k, v_k, w_k, z_k \in \mathbb{R}^n).$$

An important advantage of the modification by dyadic matrices is summarized in the following theorem:

Theorem (Sherman-Morrison-Woodbury) If A is a nonsingular matrix such that $1 + v^T A^{-1} u \neq 0$, then

$$\left(A + uv^T\right)^{-1} = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1 + v^T A^{-1} u}.$$

Matrices B_k are ought to meet some conditions. Let

$$F(x) \approx L_k(x) = F(x_k) + B_k(x - x_k), \quad B_k \approx F'(x_k)$$

be a linear approximation of function F. Iteration x_{k+1} is obtained by solving the system of equations $L_k(x) = 0$. In the case of quasi-Newton methods we stipulate that

$$L_{k+1}(x_k) = F(x_k),$$
$$L_{k+1}(x_{k+1}) = F(x_{k+1}).$$

Denote by $y_k = F(x_{k+1}) - F(x_k)$. Then from the above two conditions we obtain the so-called secant equation

$$B_{k+1} s_k = y_k.$$

A general form of the rank-one quasi-Newton formula:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) z_k^T}{z_k^T s_k},$$

where $z_k \in \mathbb{R}^n$ are suitably chosen parameters. In the case of Broyden's optimal method $z_k = s_k$.

1.9. **The Broyden's method.**

Algorithm:

(1) $x_0 \approx x^*, B_0 \approx F'(x^*)$.

(2) FOR $k = 0, 1, \ldots$

(3) $B_k s_k = -F(x_k)$

(4) $x_{k+1} = x_k + s_k$

(5) $y_k = F(x_{k+1}) - F(x_k)$

(6) $B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^T}{s_k^T s_k}$

The convergence rate of the Newton method is 2. The convergence rate of the quasi-Newton methods is just super-linear, due to the construction of matrices B_k used to approximate the Jacobian matrix. For this reason, in order to achieve the accurate approximate solutions in the quasi-Newton methods generally require several steps.

However, because in the quasi-Newton methods the computational total cost of the steps is significantly less than that of the Newton method, the use of quasi-Newton methods is more advantageous.

In minimization problems any type of quasi-Newton methods can be directly used to solve the stationary equation $F(x) = \nabla f\,(x)$. Since the Jacobian matrix of the stationary equation is $F'(x) = H(x) = Hf\,(x)$, i.e. the Hessian matrix of f, it is worth to use those quasi-Newton methods in which matrix B_k is symmetric. Nowadays the BFGS (Broyden-Fletcher-Goldfarb-Shanno) method is considered the best in this category.